

#5

Attorney Docket: 225/50239
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: DIETER GROHMANN ET AL.
Serial No.: 09/919,610 Group Art Unit: 2184
Filed: AUGUST 1, 2001 Examiner: Unknown
Title: ELEMENT FOR CARRYING OUT A PROGRAM OR TEST
SEQUENCE

CLAIM FOR PRIORITY UNDER 35 U.S.C. §119

Box PATENT APPLICATION
Commissioner for Patents
Washington, D.C. 20231

August 5, 2002

Sir:

The benefit of the filing date of prior foreign application No. 100 37 396.8, filed in Germany on August 1, 2000, is hereby requested and the right of priority under 35 U.S.C. §119 is hereby claimed.

In support of this claim, filed herewith is a certified copy of the original foreign application.

Respectfully submitted,

Gary R. Edwards

Registration No. 31,824

CROWELL & MORING, LLP
Intellectual Property Group
P.O. Box 14300
Washington, DC 20044-4300
Telephone No.: (202) 624-2500
Facsimile No.: (202) 628-8844
GRE:kms
(CAM 95309.186)



**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

Aktenzeichen: 100 37 396.8

Anmeldetag: 1. August 2000

Anmelder/Inhaber: DaimlerChrysler AG, Stuttgart/DE

Bezeichnung: Element zur Durchführung eines Programm- bzw.
Testablaufs

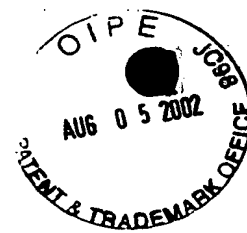
IPC: G 01 M, B 60 R, G 06 F

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 13. August 2001
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

RECEIVED
AUG 0 7 2002
OFFICE OF PETITIONS



DaimlerChrysler AG
Stuttgart

FTP/E bk
27.07.2000

Element zur Durchführung eines Programm- bzw. Testablaufs

Die Erfindung betrifft ein Element zur Durchführung und Dokumentation eines Programm- bzw. Testablaufs nach dem Oberbegriff des Patentanspruchs 1.

Technische Systeme, wie sie unter anderem in Fahrzeugen zum Einsatz kommen, sind durch eine Reihe von Eigenschaften charakterisiert. Die Systeme sind komplex. Das bedeutet, dass diese Systeme aus interagierenden Komponenten bestehen, wobei die Interaktion über Kommunikationsverbindungen wie beispielsweise Bussysteme, stattfindet. Weiterhin sind diese Systeme zumindest in ihrem Zusammenwirken heterogen. Es sind Regelungssysteme vorhanden ebenso wie reaktive Systeme. Ebenso gibt es Kombinationen beider Typen von Systemen. Weiterhin sind diese Systeme hybrid. Es kommen also elektrische, mechanische, hydraulische optische und/oder pneumatische Komponenten vor.

Die Steuerung und Regelung der Systeme wird durch spezielle Steuerkomponenten realisiert, welche Bestandteil der Systeme sind. Diese Komponenten werden im Fahrzeug als Steuergeräte bezeichnet. Diese sind durchweg elektronische Komponenten, deren Funktion in einem auf einer Recheneinheit ausgeführten Programm codiert ist.

Diese komplexen technischen Systeme müssen im Rahmen der Qualitätssicherung getestet werden. Dies gilt insbesondere für die Steuergeräte, welche letztlich die Kontrolle über die technischen Prozesse innehaben. Die hierbei verwendeten Programm- oder Testsysteme sollen die Möglichkeit bieten, auf alle für den Test relevanten Größen an der Schnittstelle der Steuerkom-

ponente bzw. Steuerkomponenten Einfluss zu nehmen und diese bei Bedarf auch sichtbar machen zu können (Test, Programmablauf, Simulation und Erfassung).

Beim Test des Steuergerätes kommt in frühen Phasen der Entwicklung eine Simulation des Steuergeräts in einer simulierten technischen Umgebung (Software-in-the-Loop, SIL) zum Einsatz. Sobald die Steuerkomponenten real verfügbar ist, wird der Test der kompletten Komponente auf einem Hardware-in-the-Loop (HIL) Testsystem fortgesetzt. An solchen HIL-Testsystemen werden auch die zum vollständigen Steuersystem gehörenden Komponenten im Verbund getestet. Im letzten Schritt werden die Steuergeräte gemeinsam mit den übrigen Komponenten zum Gesamtsystem integriert und erneut getestet.

Der vorliegenden Erfindung liegt die Aufgabe zugrunde, den Test derartiger Steuergeräte zu vereinfachen und besser handhabbar zu machen.

Diese Aufgabe wird nach der vorliegenden Erfindung gelöst durch die Verwendung von Elementen zur Durchführung einer Simulation, wobei durch das Element bestimmte Funktionen ausführbar sind, wobei das Element wenigstens einen Steuereingang aufweist, dem ein externes Steuersignal zuführbar ist, wobei in dem Element eine Variable abhängig von dem externen Steuersignal und der Ausführung der Funktion durch das Element veränderbar ist derart, dass die Variable einen bestimmten Wert ungleich "0" annimmt, wenn die Variable den Wert "0" hat und das externe Steuersignal einen ersten Signalpegel aufweist, dass die Variable weiterhin den bestimmten Wert beibehält, wenn die Variable den bestimmten Wert aufweist und erneut ein externes Steuersignal angelegt wird, das den ersten Signalpegel aufweist.

Es zeigt sich, dass sich mit solchen Elementen eine Simulation und ein Test einfach durchführen lässt. Diese Elemente können einfach mit entsprechenden Parametern versehen werden, so dass

diese dann an die jeweiligen Spezialfälle angepasst werden können.

Durch eine Verknüpfung derartiger Elemente lassen sich auch komplexe Systeme simulieren, dokumentieren, ausführen und/oder testen. Dazu werden vorteilhaft verschiedene Elemente mit verschiedenen Funktion vorgesehen. Es hat sich gezeigt, dass diese Elemente der Zahl nach begrenzt sein können, wobei sich dennoch mit vergleichsweise geringem Aufwand auch komplexe Simulationen bewältigen lassen.

Bei der Ausgestaltung des Elements nach Anspruch 2 wird die Variable auf den Wert "0" zurückgesetzt, wenn die Ausführung der Funktion durch das Element beendet wurde, wobei bei einem Rücksetzen der Variablen von dem Element über wenigstens einen Steuerausgang des Elements ein externes Steuersignal ausgegeben wird, das den ersten Signalpegel aufweist.

Dadurch kann vorteilhaft eine serielle Verknüpfung von Elementen realisiert werden. Die einzelnen Elemente können dann über die Ausgabe entsprechender externer Steuersignale die weitere Bearbeitung veranlassen, wenn sie ihre eigene Funktion beendet haben.

Bei der Ausgestaltung nach Anspruch 3 nimmt die Variable den Wert "0" an, wenn dem wenigstens einen Steuereingang des Elements ein externes Steuersignal zugeführt wird, das einen zweiten Signalpegel aufweist, wobei dann über den wenigstens einen Steuerausgang des Elements ein externes Steuersignal ausgegeben wird, das den zweiten Signalpegel aufweist.

Dadurch kann eine Deaktivierung von dem Element weitergegeben werden.

Insgesamt bedeutet dies also, dass das Element in einem aktivierten Zustand verbleiben kann und dabei die Funktion ausführen kann bis eine Deaktivierung durch ein entsprechendes exter-

nes Steuersignal erfolgt. Andererseits kann sich das Element selbst deaktivieren, wenn dessen Funktion erfüllt ist.

Vorteilhaft besteht keine Beschränkung auf die Ausführung lediglich eines Elements zu einem Zeitpunkt. Dadurch können vorteilhaft auch parallel laufende Prozesse simuliert werden.

Bei der Ausgestaltung nach Anspruch 4 weist das Element einen Steuereingang und mehrere Steuerausgänge auf, wobei das externe Steuersignal von dem Element über alle Steuerausgänge ausgegeben wird.

Mit einem solchen Element ist vorteilhaft die Ausführung der Simulation parallel laufender Prozesse initialisierbar.

Bei der Ausgestaltung des Elements nach Anspruch 5 weist das Element mehrere Steuereingänge auf, wobei ein externes Steuersignal, das den ersten Signalpegel aufweist, über den wenigstens einen Steuerausgang ausgegeben wird, wenn an allen Steuereingängen ein Steuersignal anliegt, das dem ersten Signalpegel entspricht.

Mit diesem Element lässt sich eine Aktivierung eines nachfolgenden Elementes, das über den Steuerausgang dieses Elements angestoßen wird, so steuern, dass diese Aktivierung nur dann erfolgt, wenn an allen Steuereingängen ein Aktivierungssignal anliegt. Es handelt sich also um eine logische UND-Verknüpfung.

Das Element nach Anspruch 6 weist mehrere Steuereingänge auf, wobei ein externes Steuersignal, das den ersten Signalpegel aufweist, über den wenigstens einen Steuerausgang ausgegeben wird, wenn an wenigstens einem Steuereingang ein Steuersignal anliegt, das dem ersten Signalpegel entspricht.

Im Unterschied zur Ausgestaltung nach Anspruch 5 weist die Ausgestaltung nach Anspruch 6 eine logische ODER-Verknüpfung auf.

Bei der Ausgestaltung nach Anspruch 7 weist das Element einen Steuereingang und mehrere Steuerausgänge sowie wenigstens einen Dateneingang auf, wobei ein an dem Steuereingang anliegendes externes Steuersignal mit dem ersten Signalpegel über den Steuerausgang ausgegeben wird, der abhängig von dem an dem wenigstens einen Dateneingang anliegenden Signal durch die Funktion des Elements ermittelt wird.

Vorteilhaft lässt sich dadurch eine Verzweigung des weiteren Programmablaufs, insbesondere eines Testablaufs, abhängig von bestimmten Zuständen und Datenwerten erreichen.

Bei der Ausgestaltung nach Anspruch 8 weist das Element einen oder mehrere Dateneingänge sowie einen oder mehrere Datenausgänge auf, wobei die Funktion darin besteht, aus den an dem bzw. den Dateneingängen liegenden Dateneingangssignalen ein oder mehrere Datenausgangssignale zu bilden, das bzw. die über den bzw. die Datenausgänge ausgegeben werden.

Dabei können die Dateneingangssignale im einfachsten Falle unverändert über die Datenausgänge weitergegeben werden.

Ebenso können diese Dateneingangssignale auch modifiziert werden. Dies kann beispielsweise durch arithmetische Operationen wie die Addition, Subtraktion, Multiplikation und Division erfolgen. Ebenso ist es auch möglich, den Absolutwert zu bilden, eine Integration oder Differentiation vorzunehmen oder transzendente Funktionen zu realisieren.

Ebenso können die Funktion auch darin bestehen, die Dateneingangssignale mittels Boolescher Algebra zu bewerten durch NOT-, AND-, OR- sowie XOR-Verknüpfungen.

Es ist auch möglich, dass die Funktion einen Vergleich darstellt. Dieser Vergleich kann so ausgestaltet sein, dass mehrere Dateneingangssignale auf ihren Wert hin verglichen werden. Ebenso können ein oder mehrere Dateneingangssignale daraufhin

untersucht werden, ob sie in einem bestimmten Wertebereich liegen. Weiterhin kann auch die Änderung des Dateneingangssignals untersucht werden, indem bei dem Vergleich beispielsweise untersucht wird, ob das jeweilige Dateneingangssignal eine Schwelle von unten oder von oben überschreitet. Vorteilhaft wird von einem solchen vergleichenden Element kein externes Steuersignal ausgegeben. Das Element führt die vergleichende Funktion aus, nachdem es über einen Steuereingang ein externes Steuersignal mit dem ersten Signalpegel empfangen hat. Der Vergleich wird dann beendet, wenn über den Steuereingang eine Deaktivierung über ein entsprechendes externes Steuersignal erfolgt.

Die Funktion kann auch darin bestehen, bestimmte Werte zu speichern. Dies kann beispielsweise der letzte Wert des Dateneingangssignals sein, der kleinste Wert, den das Dateneingangssignal angenommen hat, den größten Wert, den das Dateneingangssignal angenommen hat oder den Mittelwert des Dateneingangssignals.

Bei der Ausgestaltung nach Anspruch 9 besteht die Funktion des Elements in einer Zeitmessung.

Das Element kann beispielsweise wenigstens einen Dateneingang aufweisen, wobei nach der Aktivierung des Elements durch ein externes Steuersignal die Zeitdauer bestimmt wird, die vergeht, bis nach einem bestimmten ersten Dateneingangssignal ein bestimmtes weiteres Dateneingangssignal vorliegt.

Bei der Ausgestaltung nach Anspruch 10 erfolgt die Zeitmessung, indem beginnend zu einem Zeitpunkt, der durch das Anliegen eines externen Steuersignals mit dem ersten Signalpegel oder durch ein bestimmtes Signal an einem Dateneingang bzw. eine Signalkombination an mehreren Dateneingängen des Elements definiert ist, eine bestimmte Zeitdauer gemessen wird, wobei am Ende der Zeitdauer über einen Steuerausgang des Elements ein externes Steuersignal mit dem ersten Signalpegel ausgegeben wird

und/oder an einem oder mehreren Datenausgängen ein entsprechendes Datensignal ausgegeben wird.

Eine solche Ausgestaltung stellt einen zeitbezogenen Count-Down, insbesondere ein Time-Out-Glied, dar.

Bei der Ausgestaltung des Elementes nach Anspruch 11 ist dem Element über einen Steuereingang ein externes Steuersignal zuführbar, wobei über einen Datenausgang des Elements ein zeitlicher Signalverlauf als Datenausgangssignal ausgegeben wird, wenn das externe Steuersignal den ersten Signalpegel aufweist.

Dabei sind verschiedene Funktionen durch eine entsprechende Verknüpfung von Basisfunktionen realisierbar.

Vorteilhaft zeigt sich, dass mit den beschriebenen Elementen Testsystemen realisierbar sind, die einen automatisierten Ablauf oder Test gestatten. Insbesondere ist es also nicht mehr notwendig, die Vielzahl und Komplexität der Steuerfunktionen durch ein manuelles Testen zu überprüfen, was ohnehin unter dem Gesichtspunkt der Effektivität und Effizienz problematisch ist.

Mit den Elementen wird sowohl die Simulation als auch die Erfassung und Auswertung von für den Test oder die Software relevanten Größen gestattet. Dies erfolgt durch eine Simulation der im Test befindlichen Komponente oder des im Test befindlichen Systems sowie eine nachfolgende Auswertung der entsprechenden Reaktionen.

Weiterhin zeigt sich, dass die Simulation und Auswertung in Echtzeit möglich ist. Weiterhin zeigt sich, dass ein Programm- oder Testsystem, das aus den beschriebenen Elementen besteht, auch in der Lage ist, mehrere Funktionen gleichzeitig auszuführen. Weiterhin ist durch eine entsprechende Ausgestaltung der Elemente sowohl die Verarbeitung digitaler Größen wie auch die Verarbeitung analoger Größen möglich.

Aus dem Stand der Technik sind im Zusammenhang mit der computergestützten Simulation zwei Verfahren bekannt. Das eine betrifft die Kompilierung. Diese erfordert eine Übersetzung des Programms vor der Laufzeit in eine maschinenlesbare Form. Diese maschinenlesbare Form ist dann wiederum nur auf der Maschine ausführbar, für die diese maschinenlesbare Form kompiliert wurde. Weiterhin ist die Interpretation bekannt. Dabei wird das Programm während der Laufzeit analysiert und ausgeführt. Dies hat Nachteile hinsichtlich der erreichbaren Laufzeiten.

Durch die vorliegend vorgeschlagene Zusammenstellung der Elemente wird erreicht, dass die Elemente durch Vorgabe entsprechender Parameter an die konkreten Verhältnisse anpassbar sind. Dadurch wird zum einen eine gute Flexibilität erreicht. Dennoch sind die Laufzeiten der Simulationen hinreichend schnell und auch flexibel auf verschiedenen Rechnern ausführbar.

Ein Ausführungsbeispiel der Erfindung ist in der Zeichnung dargestellt. Es zeigt dabei im einzelnen:

Figuren 1 bis 17: verschiedene Elemente, durch deren Kombination und Verknüpfung verschiedene Simulations- und Testaufgaben lösbar sind,

Figuren 18 und 19: Darstellungen von Elementen zur Realisierung von zusammengesetzten Funktionen,

Figur 20: eine Darstellung eines datenspeichernden Elements,

Figur 21: eine Darstellung zur Erzeugung einer Liste, mit der die Elemente verknüpfbar sind und

Figur 22: ein Beispiel für eine Verknüpfung zur Durchführung einer Simulation.

In dem Gesamtsystem wird zwischen steuernden und speichernden Elementen unterschieden.

Steuernde Elemente 1 erfüllen eine klar umrissene Funktion. Der grundlegende Aufbau eines steuernden Elementes 1 ist in Figur 1

dargestellt. Dem steuernden Element 1 wird als Kontrollfluss ein externes Steuersignal 2 über einen Steuereingang 3 zugeführt. Weiterhin ist ein Steuerausgang 4 zu sehen, über den von dem steuernden Element 1 ein externes Steuersignal 5 ausgebar ist.

Weiterhin sind Dateneingänge 6 und 7 zu sehen, über die Dateneingangssignale 8 und 9 zuführbar sind.

Mit der Bezugsziffer 10 ist eine Variable bezeichnet, die entsprechend dem aktivierten oder deaktivierten Zustand des Elements 1 einen bestimmten Wert annimmt, der ungleich 0 ist, oder die den Wert 0 aufweist, wenn das Element 1 deaktiviert ist.

Mit der Bezugsziffer 11 ist einer oder auch mehrere Parameter bezeichnet, die dem steuernden Element 1 vorgebar sind und auf die das steuernde Element 1 während der Abarbeitung seiner Funktion zugreifen kann. Diese Parameter sind während der Aktivierungsdauer eines steuernden Elements 1 nicht veränderbar.

Das steuernde Element 1 weist weiterhin Datenausgänge 12 und 13 auf, über die Datenausgangssignale 14 und 15 ausgebar sind. Über diese Datenausgänge werden die verarbeiteten Daten anderen Elementen zur Verfügung gestellt.

Diese Datenflüsse transportieren also Werte oder Ereignisse, das heißt Datensignale, die bestimmte Zustände repräsentieren. Die von den Datenflüssen transportierten Größen können sich über der Zeit ändern.

Zur Erfüllung komplexer Aufgaben können steuernde Elemente miteinander verkettet werden. Die Verkettung kann sowohl über die Kontrollflüsse, das heißt also über die externen Steuersignale an den Steuereingängen sowie Steuerausgängen, als auch über die Datenflüsse, das heißt über die Dateneingangs- und Datenausgangssignale, erfolgen.

Die steuernden Elemente lassen sich in zwei Gruppen unterteilen:

- Kontrollierende Elemente: Diese überwachen die Weiterleitung externer Steuersignale und damit die Aktivierung und Deaktivierung von steuernden Elementen sowie
- Datenverarbeitende Elemente: Diese Elemente simulieren und überwachen Größen, die von dem Testsystem bereitgestellt werden.

Die Ablaufsteuerung erfolgt über die externen Steuersignale 2, 5 sowie über die Variable 10. Jedes steuernde Element 1 verfügt über genau eine Variable 10. Diese zeigt den Zustand der Aktivierung und Deaktivierung an wie oben beschrieben. Wird dem steuernden Element 1 über den Steuereingang 3 ein externes Steuersignal mit dem ersten Signalpegel zugeführt, wird die Variable 10 auf den Wert ungleich 0 gesetzt, wenn diese vorher den Wert 0 hatte. War der Wert der Variablen 10 bereits ungleich 0, wird dieser Wert beibehalten.

Solange die Variable 10 den Wert ungleich 0 aufweist, wird von dem steuernden Element 1 dessen Funktion abgearbeitet. Wird während dem Abarbeiten der Funktion über den Steuereingang 3 wiederum ein externes Steuersignal 2 mit dem ersten Signalpegel zugeführt, wird das Abarbeiten der Funktion fortgesetzt und nicht abgebrochen und neu begonnen.

Wenn die Funktion des steuernden Elements 1 abgearbeitet ist, wird die Variable 10 auf den Wert 0 gesetzt. Das heißt, dass dieses steuernde Element 1 deaktiviert wird. Über den Steuerausgang 4 wird ein externes Steuersignal 5 ausgegeben, das den zweiten Signalpegel aufweist und zu einer Deaktivierung der steuernden Elemente führt, an deren Steuereingängen dieses externe Steuersignal anliegt.

Wird einem steuernden Element 1 über seinen Steuereingang 3 ein externes Steuersignal 2 zugeführt, das den zweiten Signalpegel aufweist, wird die Variable 10 auf den Wert 0 gesetzt, wenn

diese vorher einen Wert ungleich 0 hatte. Die Abarbeitung der Funktion wird dann abgebrochen. Unabhängig von dem vorherigen Wert der Variablen 10 wird über den Steuerausgang 4 ein externes Steuersignal 5 ausgegeben, das den zweiten Signalpegel aufweist und zur Deaktivierung der steuernden Elemente führt, an deren Steuereingängen dieses externe Steuersignal liegt.

Diese steuernden Elemente können seriell angeordnet werden oder auch parallel. In diesem Fall können die Variablen mehrerer steuernder Elemente einen Wert ungleich 0 aufweisen. Deren Funktionen werden dann parallel abgearbeitet.

Figur 2 zeigt ein Start-Element 201. Dieses Start-Element 201 weist keinen Steuereingang auf sondern nur einen Steuerausgang 204, über den ein externes Steuersignal 205 ausgebar ist. Sofern dieses externe Steuersignal 205 den ersten Signalpegel aufweist, lassen sich dadurch andere steuernde Elemente aktivieren, denen dieses externe Steuersignal an einem Steuereingang zugeführt wird.

Alternativ zur Darstellung der Figur 2 kann das Start-Element 201 auch mit einem Steuereingang versehen sein. Nachdem dann über das Start-Element 201 ein externes Steuersignal 205 mit dem ersten Signalpegel ausgegeben wurde, kann über diesen Steuereingang das Start-Element veranlasst werden, ein externes Steuersignal 205 mit dem zweiten Signalpegel auszugeben und dadurch die Aktivierung aller steuernden Elemente definiert zu beenden, indem von diesen wiederum dieses externe Steuersignal mit dem zweiten Signalpegel weitergegeben wird.

Figur 3 zeigt ein Terminator-Element 301. Mit einem solchen Terminator-Element 301 wird ein Kontrollfluss beendet. Das bedeutet, das dieses Terminator-Element 301 einen externen Steuereingang 303 aufweist, dem ein externes Steuersignal 302 zugeführbar ist. Ein externes Steuersignal 302 mit einem ersten Signalpegel wird von diesem Terminator-Element "geschluckt", ohne

dass dieses externe Steuersignal an andere steuernde Elemente weitergegeben wird.

Figur 4 zeigt ein Exit-Element 401. Mit einem solchen Exit-Element 401 wird ein Kontrollfluss beendet. Im Unterschied zu einem Terminator-Element 301 entsprechend der Darstellung der Figur 3 ist dieses Exit-Element 401 allerdings mit einem Start-Element verbunden. Wird dem Exit-Element 401 über seinen Steuereingang 403 ein externes Steuersignal 402 zugeführt, das den ersten Signalpegel aufweist, wird von diesem Exit-Element über dessen Steuerausgang 404 ein externes Steuersignal 405 entsprechend dem zweiten Signalpegel an das Start-Element ausgegeben. Von diesem Start-Element wird dieses externe Steuersignal mit dem zweiten Signalpegel sofort weitergegeben, so dass dadurch alle steuernden Elemente deaktiviert werden. Ein Exit-Element 401 beendet somit einen durch ein Start-Element initiierten Prozess. Wird dem Exit-Element 401 ein externes Steuersignal mit dem zweiten Signalpegel zugeführt, wird dieses externe Steuersignal ignoriert.

Figur 5 zeigt ein Fork-Element 501, mit dem ein über einen Steuereingang 503 eingehendes externes Steuersignal 502 über mehrere Steuerausgänge 504 ausgegeben wird. Dadurch können mit diesem Fork-Element 501 gleichzeitig mehrere andere steuernde Elemente aktiviert oder deaktiviert werden.

Figur 6 zeigt ein Joint-Element 601. Dieses Joint-Element 601 weist mehrere Steuereingänge 603 auf, denen jeweils ein externes Steuersignal 602 zuführbar ist. Daraus wird ein externes Steuersignal 605 abgeleitet, das über den Steuerausgang 604 ausgegeben wird.

Weiterhin ist der Darstellung nach Figur 6 zu entnehmen, dass mittels eines Parameters 611 vorgebar ist, wie das externe Steuersignal 605 aus den eingehenden externen Steuersignalen 602 abgeleitet wird.

Dies kann beispielsweise über eine logische UND-Verknüpfung der externen Steuersignale 602 an den Steuereingängen 603 erfolgen. Das bedeutet, dass erst dann ein externes Steuersignal 605 mit dem ersten Signalpegel über den Steuerausgang 604 ausgegeben wird, wenn an jedem Steuereingang wenigstens einmal ein externes Steuersignal 602 angelegen hat mit dem ersten Signalpegel.

Ebenso kann das externe Steuersignal 605, das über den Steuerausgang 604 ausgegeben wird, auch über eine logische ODER-Verknüpfung realisiert sein. Das bedeutet dann, dass über den Steuerausgang 604 sofort dann ein externes Steuersignal 605 mit dem ersten Signalpegel ausgegeben wird, wenn an wenigstens einem der Steuereingänge 603 ein externes Steuersignal 603 anliegt, das den ersten Signalpegel aufweist.

Ebenso gilt dies für die Weitergabe von externen Steuersignalen mit dem zweiten Signalpegel.

Figur 7 zeigt ein Verzweigungs-Element 701. Diesem Verzweigungselement 701 ist über einen Steuereingang 703 ein externes Steuersignal 702 zuführbar. Wenn dieses Steuersignal 702 den ersten Signalpegel aufweist, wird die Variable 710 auf den entsprechenden Wert gesetzt und das Verzweigungs-Element 701 wird aktiviert. Abhängig von dem an einem Dateneingang 707 anliegenden Dateneingangssignal 708 wird ermittelt, über welchen der externen Steuerausgänge 704 das Steuersignal 705 mit dem ersten Signalpegel ausgegeben wird.

Die Überprüfung abhängig von dem Dateneingangssignal 708 kann beispielsweise als "if-then-else"-Überprüfung mit zwei Steuerausgängen 704 realisiert sein. Ebenso ist es möglich, mehr als zwei Steuerausgänge 704 vorzusehen und die Überprüfung abhängig von dem Dateneingangssignal 708 als "Select"-Überprüfung vorzunehmen. Bei dieser Select-Überprüfung wird dann ermittelt, über welchen der Steuerausgänge 704 das externe Steuersignal 705 mit dem ersten Signalpegel ausgegeben wird.

Figur 801 zeigt ein Steuerelement 801, dessen Funktion in der Abarbeitung einer Schleife besteht. Diese Schleife besteht aus einem Schleifenkörper und einer Schleifenbedingung. Solange die Schleifenbedingung erfüllt ist, wird der Schleifenkörper ausgeführt. Eine Ausführung des Schleifenkörpers von der Aktivierung bis zu dessen Beendigung wird als Durchlauf bezeichnet. Ist die Bedingung nicht erfüllt, wird die Schleife inaktiv. Die Bedingung wird nur zu bestimmten Zeiten geprüft. Entsprechend dem Zeitpunkt der Überprüfung können bestimmte Schleifentypen unterschieden werden.

Bei der sogenannten unbedingten Schleife erfolgt eine definierte Anzahl von Schleifendurchläufen. Die Schleifenbedingung ist erfüllt, solange die definierte Anzahl der Durchläufe noch nicht erreicht ist. Sie wird jeweils vor einem Durchlauf geprüft. Für die Entscheidung, ob die Schleife durchlaufen wird oder nicht, werden keine weiteren externen Daten benötigt. Deswegen ist das Dateneingangssignal 808 sowie der Dateneingang 807 in der Figur 8 strichliniert dargestellt, da diese bei der unbedingten Schleife allenfalls dann benötigt werden, wenn während der Verarbeitung im Schleifendurchlauf auf externe Daten zugegriffen werden muss.

Es sind auch sogenannte bedingte Schleifen bekannt. Bei der annehmenden Schleife wird der Schleifenkörper einmal durchlaufen bevor die Schleifenbedingung überprüft wird. Die Schleifenbedingung wird dann durch die Auswertung wenigstens eines Dateneingangssignals 808 überprüft. Bei der abweisenden Schleife wird die Schleifenbedingung bereits vor dem ersten Durchlauf geprüft. Ist diese Bedingung zum Zeitpunkt der Aktivierung der Schleife nicht erfüllt, findet überhaupt kein Durchlauf durch die Schleife statt.

Neben diesen Elementen, die in erster Linie den Kontrollfluss steuern, indem entsprechende Steuersignale zur Aktivierung und Deaktivierung anderer Elemente weitergegeben werden, sind auch datenverarbeitende Elemente vorgesehen. Bei diesen Elementen

sind je nach spezieller Art der Datenverarbeitung Dateneingänge und Datenausgänge zur Zuführung von Dateneingangssignalen und zur Abführung von Datenausgangssignalen vorgesehen.

Die Daten können zeitkontinuierliche Größen sein. Daten dieses Typs verfügen zu jedem Zeitpunkt über einen Wert.

Weiterhin können die Daten zeitdiskrete Größen sein. Daten dieses Typs haben keinen Wert im engeren Sinne sondern stellen ein Ereignis in der Zeit dar. Zeitdiskrete Größen werden daher kurz als Ereignis (Event) bezeichnet.

Figur 9 zeigt ein datenverarbeitendes Element 901. Dieses Element 901 weist zwei Dateneingänge 907 auf, denen jeweils ein Dateneingangssignal 908 zugeführt wird. Weiterhin weist dieses Element 901 zwei Datenausgänge 913 auf, über die jeweils ein Datenausgangssignal 915 ausgebbar ist.

Dieses Element 901 kann als Datenverarbeitung beispielsweise eine einfache Zuweisung ausführen. Dabei wird ein eingehendes Dateneingangssignal 908 über wenigstens einen Datenausgang 913 als Datenausgangssignal 915 weitergegeben. Es ist dabei möglich, dass das Element 901 lediglich einen Dateneingang sowie lediglich einen Datenausgang aufweist.

Weiterhin kann das Element 901 als modifizierendes Element ausgebildet sein. Als modifizierendes Element verarbeitet dieses Element eingehende Dateneingangssignale 908 zu einer oder mehreren Ausgangsgrößen, die als Datenausgangssignale 915 über entsprechende Datenausgänge 913 ausgegeben werden. Diese modifizierenden Elemente können beispielsweise mathematische Operationen ausführen. Diese mathematischen Operationen können beispielsweise arithmetische Operationen sein wie die Addition, die Subtraktion, die Multiplikation und die Division. Weiterhin können auch andere Operatoren realisiert sein wie beispielsweise Integratoren, Differentiatoren oder transzendente Funktio-

nen. Ebenso kann beispielsweise der Absolutwert gebildet werden.

Diese Elemente 901 beenden die Durchführung ihrer Funktion, wenn ihnen ein externes Steuersignal 902 mit dem zweiten Signalpegel über den Steuereingang 903 zugeführt wird.

Weiterhin ist es möglich, mittels der datenverarbeitenden Elemente 901 Boolesche Operatoren auszuführen. Diese Operatoren der Booleschen Algebra können beispielsweise die NOT-, AND-, OR sowie die XOR-Funktion sein.

Ebenso können vergleichende Elemente realisiert werden. Diese vergleichen zwei oder mehr Eingangsgrößen 908 und stellen das Ergebnis des Vergleichs am Datenausgang 913 mittels eines entsprechenden Datenausgangssignals 915. Vergleichende Elemente verfügen vorteilhaft über einen Ereignisausgang, welcher ein Ereignis generiert, sobald der Vergleich wahr wird. Es sind beispielsweise nachfolgende Vergleiche möglich:

- Äquivalenz: Die eingehenden Größen weisen denselben Wert auf
- Antivalenz: Die eingehenden Größen weisen unterschiedliche Werte auf.
- Größer: Der erste Wert ist größer als der zweite.
- Größer oder gleich: Der erste Wert ist größer als der zweite oder gleich dem zweiten.
- Kleiner: Der erste Wert ist kleiner als der zweite.
- Kleiner oder gleich: Der erste Wert ist kleiner als der zweite oder gleich dem zweiten.
- Innerhalb: Der eingehende Wert liegt zwischen einer oberen und einer unteren Grenze.
- Außerhalb: Der eingehende Wert liegt außerhalb einer durch eine untere und eine obere Grenze spezifizierten Bereichs.
- Steigende Flanke: Der eingehende Wert überschreitet eine Schwelle von unten nach oben.

- Fallende Flanke: Der eingehende Wert unterschreitet eine Schwelle von oben nach unten.

Weiterhin können datenverarbeitende Elemente als haltende Elemente realisiert sein. Diese haltenden Elemente verfügen über je einen Ereignis- und einen Werteingang. Sie verarbeiten den Wert, sobald an ihrem Ereigniseingang ein Ereignis vorliegt. Folgende Verarbeitungen haltender Elemente können realisiert sein:

- Buffer: Der Puffer speichert den Wert des eingehenden Wertes, sooft ein Ereignis am Ereigniseingang vorliegt.
- Minimalwertspeicher: Der Minimalwertspeicher enthält den kleinsten beobachteten Wert.
- Maximalwertspeicher: Der Maximalwertspeicher enthält den größten beobachteten Wert.
- Mittelwertspeicher: Der Mittelwertspeicher enthält den Mittelwert des beobachteten Wertes.

Für die Minimalwertspeicher, Maximalwertspeicher und Mittelwertspeicher kann vorteilhaft noch festgelegt werden, ob die Speicherung über alle Werte, die letzten n Werte oder die letzten t Sekunden erfolgen soll.

Gemäß der Darstellung der Figur 10 können auch zeitbezogene Elemente 1001 vorhanden sein.

Ein solches zeitbezogenes Element 1001 kann beispielsweise als Timeout realisiert sein. Dabei läuft eine voreingestellte Zeit t ab. Nach Ablauf der Zeit t wird ein Ereignis ausgelöst und das Element 1001 wird deaktiviert unter Abgabe eines externen Steuersignals 1005 des zweiten Pegels. Über die Datenausgänge 1013 können wertbezogene und/oder ereignisbezogene Datenausgangssignale 1015 ausgegeben werden. Ebenso ist es möglich, nur einen Datenausgang 1013 vorzusehen.

Figur 11 zeigt ein zeitbezogenes Element 1101, das als Timer arbeitet. Mit dem zeitbezogenen Element 1101 wird die Zeit zwi-

schen zwei Ereignissen gemessen. Die Messung beginnt mit einem Ereignis, das als Dateneingangssignal 1108 über einen der Dateneingänge 1107 zugeführt wird. Die Zeitmessung endet mit einem weiteren Ereignis, das dem Element 1101 an dem anderen Dateneingang 1107 als Dateneingangssignal 1108 zugeführt wird. Ebenso ist es auch möglich, nur einen Dateneingang 1107 vorzusehen und die beiden Datensignale 1108, die den Anfang und das Ende des zu messenden Zeitintervalls definieren, über einen Dateneingang 1107 zuzuführen.

Der Timer kann beispielsweise zwei Datenausgänge 1113 aufweisen. An einem wertbezogenen Datenausgang kann die gemessene Zeit als Datenausgangssignal 1115 ausgegeben werden. Weiterhin kann das Element 1101 ein Ereignis generieren und über einen weiteren Datenausgang 1113 als Datenausgangssignal 1115 ausgeben. Nach Abschluss der Messung wird das Element 1101 deaktiviert unter Abgabe eines externen Steuersignals 1105 mit dem zweiten Signalpegel.

Weiterhin können ereignisbezogene Elemente vorgesehen werden. Diese Elemente reagieren auf Ereignisse.

Ein Beispiel eines solchen ereignisbezogenen Elements ist in Figur 12 dargestellt. Figur 12 zeigt ein Countdown-Element 1201. Das Countdown-Element 1201 zählt bei jedem über den Dateneingang 1207 eintreffenden Ereignis, das durch das Dateneingangssignal 1208 repräsentiert wird, einen voreingestellten Zähler von dem Wert n auf den Wert 0 herunter. Wird der Zählerstand 0 erreicht, so löst das Countdown-Element 1201 seinerseits ein Ereignis aus, das über einen der beiden dargestellten Datenausgänge 1213 als Datenausgangssignal 1215 ausgegeben wird. Weiterhin wird das Countdown-Element 1201 deaktiviert, wobei weiterhin ein externes Steuersignal mit dem ersten Signalpegel über den Steuerausgang 1204 als externes Steuersignal 1205 ausgegeben wird. Der Wert n kann beispielsweise über den Parameter 1211 voreingestellt werden.

Abweichend von der Darstellung der Figur 12 ist es auch möglich, nur einen Datenausgang 1213 vorzusehen.

Figur 13 zeigt ein Counter-Element. Nach dessen Aktivierung über ein externes Steuersignal 1302 an dessen Steuereingang 1303 zählt dieses Element die Ereignisse, die dem Counter-Element 1302 an dessen Dateneingang 1307 als Dateneingangssignal 1308 zugeführt werden. Die Zahl der Ereignisse wird dann über die Datenausgänge 1313 als Datenausgangssignal 1315 ausgegeben. Das Counter-Element 1301 zählt die Ereignisse nach dessen Aktivierung über das externe Steuersignal 1302 mit dem ersten Signalpegel bis dem Counter-Element 1301 ein externes Steuersignal 1302 mit dem zweiten Signalpegel zugeführt wird.

Auch hierbei ist es wieder möglich, lediglich einen Datenausgang 1313 vorzusehen.

Figur 14 zeigt ein Guardian-Element 1401. Dieses Guardian-Element 1401 überwacht zwei Dateneingänge 1407 darauf, an welchem der beiden Dateneingänge 1407 zuerst ein Dateneingangssignal 1408 anliegt, das ein Ereignis repräsentiert. Liegt ein solches ein Ereignis repräsentierendes Dateneingangssignal 1408 vor, wird an den Datenausgängen 1413 ein entsprechendes Datenausgangssignal 1415 ausgegeben. Dieses Datenausgangssignal 1415 ist ereignisbezogen. Das Guardian-Element 1401 erzeugt ebenfalls ein Ereignis, das als Datenausgangssignal 1415 ausgegeben wird. Anschließend wird das Guardian-Element 1401 deaktiviert, wobei über den Steuerausgang 1404 ein externes Steuersignal mit dem ersten Signalpegel ausgegeben wird. Mit einem solchen Guardian-Element 1401 lassen sich Wettläufe von zeit- und/oder ereignisbezogenen Prozessen überwachen. Auch hierbei kann wiederum lediglich ein Datenausgang 1413 vorgesehen werden.

Figuren 15 und 16 zeigen Konverter-Elemente. Mit diesen Konverter-Elementen lassen sich Steuersignale umsetzen in ereignisbezogene Datensignale und umgekehrt.

Figur 15 zeigt ein Ereignisgenerator-Element 1501. Dieses Element erzeugt an seinem Datenausgang 1513 ein ereignisbezogenes Datenausgangssignal 1515, wenn an seinem Steuereingang 1503 ein externes Steuersignal 1502 mit dem ersten Signalpegel anliegt. Dieses externe Steuersignal 1502 wird unmittelbar über den Steuerausgang 1504 als externes Steuersignal 1505 weitergegeben.

Figur 16 zeigt ein Synchronisations-Element 1601. Dieses Synchronisations-Element 1601 gibt ein externes Steuersignal 1502 mit dem ersten Signalpegel erst dann über den Steuerausgang 1604 als externes Steuersignal 1605 weiter, wenn an seinem Dateneingang 1607 ein bestimmtes ereignisbezogenes Dateneingangssignal 1608 anliegt.

Figur 17 zeigt ein Signalgenerator-Element 1701. Dies ist ein Element, mit dem Signalverläufe über der Zeit erzeugt werden können. Das Signalgenerator-Element wird deaktiviert, wenn der Signalverlauf vollständig abgespielt ist oder wenn dem Signalgenerator-Element 1701 über seinen Steuereingang 1703 ein externes Steuersignal 1702 mit dem zweiten Signalpegel zugeführt wird. Signalverläufe werden ihrerseits wiederum durch spezialisierte Signalelemente beschrieben. es können drei Signalelemente unterschieden werden:

- Ein Signalverlauf besteht aus verschiedenen Segmenten, die zeitlich aufeinander folgen.
- Ein Segment wird durch ein Verknüpfungselement beschrieben.
- Ein Verknüpfungselement verschaltet ein oder mehrere Signalgeneratorelemente.
- Ein Signalgeneratorelement erzeugt einen über der Zeit veränderlichen Wert.

Ein Verknüpfungselement ist durch einen oder mehrere Slots 1801 gekennzeichnet, in die Signalgenerator-Elemente 1802 eingefügt werden können, entsprechend der Darstellung der Figur 18. Die Slots 1801 sind verknüpft über eine oder mehrere Operationen,

welche die Verknüpfung definieren. So können aus elementaren Signalfunktionen komplexe Verläufe am Ausgang Y des Verknüpfungselements 1802 erzeugt werden. Es sind insbesondere Verknüpfungen für folgende Operationen definiert:

- Addition: $Y = A + B$
- Subtraktion: $Y = A - B$
- Multiplikation: $Y = A * B$
- Division: $Y = A / B$, ist hierbei $B = 0$, so nimmt Y den maximal darstellbaren positiven Wert an, wenn $A > 0$ ist und den maximal darstellbaren negativen Wert, wenn $A < 0$ ist.
- MAC: $Y = A * B + C$

Weiterhin können Filter vorgesehen sein:

- Tiefpass: Der Ausgang Y ergibt sich aus dem tiefpassgefilterten Signal A. Die Grenzfrequenz des Tiefpasses wird von Signal B vorgegeben.
- Hochpass: Der Ausgang Y ergibt sich aus dem hochpassgefilterten Signal A. Die Grenzfrequenz des Hochpasses wird von dem Signal B vorgegeben.
- Bandpass: Der Ausgang Y ergibt sich aus dem bandpassgefilterten Signal A. Die Grenzfrequenzen des Bandpasses werden von Signalen B und C vorgegeben.
- Bandsperre: Der Ausgang Y ergibt sich aus dem bandgesperrten Signal A. Die Grenzfrequenzen der Bandsperre werden von den Signalen B und C vorgegeben.

Aus den aufgeführten Beispielen ist ersichtlich, dass dem Verknüpfungselement 1802 gegebenenfalls auch mehr als zwei Größen zugeführt werden können.

Ein Signalgeneratorelement erzeugt einen zeitlich veränderlichen Wert. Hierzu kann vorteilhaft auf eine Bibliothek von vorgefertigten Generatorelementen zugegriffen werden. Diese sind parametrierbar. Folgende Elemente können vorteilhaft definiert sein:

- Sinus: $Y = A + \sin (w*t + b)$; mit der Amplitude A, der Kreisfrequenz w und der Phase b.

- Cosinus: $Y = A * \cos (w*t + b)$; mit der Amplitude A, der Kreisfrequenz w und der Phase b.
- Exponentialfunktion: $y = \exp (A*t)$
- Konstante: $Y = \text{const.}$
- Pulsetrain: Y ist eine Folge von Pulsen der Dauer T_{Pulse} und der Periodenzeit T_{Periode}
- PWM: Y ist eine Folge von Pulsen mit dem Tastverhältnis T_{Ratio} und der Periodendauer T_{Periode} .
- Rampe: Y ist eine Rampe von A nach B mit der Steigung M.
- Messdaten: Dabei wird Y nicht analytisch bestimmt sondern einer zuvor aufgenommen Messreihe entnommen. Somit ist es möglich, in der Realität gemessene Verläufe wiederzugeben.

Mit dem dargestellten Beispiel der Figur 19 lässt sich beispielsweise ein Klopfsignal eines Klopfensors einer Brennkraftmaschine nachbilden. Dieses Signal wird durch eine abklingende Sinusschwingung charakterisiert. Der Effekt kann durch die Gleichung $Y = A * \exp(-c*t) * \sin (w*t)$ beschrieben werden. Das resultierende Signal Y kann durch die Verknüpfung 1903 der Multiplikation und die Signalgeneratoren 1901 der Exponentialfunktion und 1902 der Sinusfunktion erzeugt werden.

Weiterhin können speichernde Elemente 2001 vorgesehen werden. Speichernde Elemente sind nicht in den Fluss der Steuersignale eingebunden. Sie verfügen lediglich über Datensignale. Ein Beispiel ist in Figur 20 dargestellt. Diese speichernden Elemente 2001 verfügen nicht über eine Variable entsprechend zu den anderen Elementen und weisen keinen Zustand "aktiv" oder "deaktiviert" auf.

Die in Figur 20 dargestellten speichernden Elemente 2001 können im wesentlichen in zwei Formen auftreten.

- Quellen: Quellen haben keinen Eingang und einen Ausgang. An diesem liegt die gespeicherte Größe zur weiteren Verarbeitung an.

- Senken: Senken nehmen Größen zur Speicherung auf. Die gespeicherte Größe kann über eine korrespondierende Quelle jederzeit abgerufen werden.

Quellen greifen auf einen Speicherbereich zu und stellen die dort gespeicherte Größe an ihrem Ausgang zur Verfügung.

Beispielsweise kann an dem Ausgang eine konstante Größe zur Verfügung gestellt werden, die zeitlich unveränderlich ist.

Ebenso kann eine solche Quelle auch als Signalquelle ausgebildet sein, die auf zeitlich veränderliche Signale des Programm- oder Testsystems zugreift und diese den steuernden Elementen zur Verfügung stellt. Die verfügbaren Signale sind dann durch das Programm- oder Testsystem vorgegeben.

Weiterhin kann die Quelle auch als Variablenquelle definiert sein. Variablenquellen liefern den Wert eines Variablenspeichers, der einen zeitlich veränderlichen Wert enthält. Variablen können im Unterschied zu Signalen frei definiert werden.

Senken hingegen nehmen Größen auf und speichern sie in einer Speicherstelle.

Signalensenken speichern die eingehende Größe in einem vom Programm- oder Testsystem zur Verfügung gestellten Signal. Die verfügbaren Signale sind durch das Testsystem vorgegeben.

Variablensenken speichern die eingehende Größe in einem frei definierbaren Variablenspeicher.

Den grundlegenden Aufbau eines Testsystems 2102 mit der Fähigkeit zur Testautomation zeigt Figur 21. In das Testsystem 2102 wird eine (Software-)Komponente integriert, die als Bestandteil des Testsystems 2102 für die Ausführung von Testbeschreibungen 2101 zuständig ist. Diese Komponente wird im folgenden als

Testmaschine 2103 bezeichnet. Ihr Aufbau wird im folgenden näher erläutert.

Kerngedanke der Testmaschine 2103 ist die Abbildung der vorstehend dargestellten Elemente auf entsprechende Datenstrukturen, die von einem Algorithmus benutzt werden. Basis des Algorithmus sind zwei Module, das Elementenmodul und das Managementmodul.

Bei dem Elementenmodul bildet ein Modul die Grundlage für jeweils ein Element. Dieses wird als Elementenmodul bezeichnet. Das Elementenmodul definiert Datenstrukturen für jedes darzustellende Element sowie Programmcode, der für die Initialisierung und Ausführung des jeweiligen Elementes notwendig ist. Die jeweils zu einem Element gehörigen Datenstrukturen und der dazugehörige Code werden als Elementklasse bezeichnet. Eine mit konkreten Daten versehene Datenstruktur zu einem Element wird als Elementobjekt bezeichnet. Der Vorgang der Erzeugung eines neuen Elementobjekts wird als Instanziierung eines Elementobjekts bezeichnet.

Das Managementmodul verwaltet alle Elementenmodule. Im Managementmodul definierte Datenstrukturen mit dazugehörigem Code werden als Managementklasse bezeichnet. Eine mit konkreten Daten gefüllte Datenstruktur einer Managementklasse wird mit Managementobjekt bezeichnet. Der Vorgang der Erzeugung eines neuen Managementobjekts wird als Instanziierung eines Managementobjektes bezeichnet.

Die Elementklassen des Elementenmoduls und die Managementklassen des Managementmoduls sollen nachfolgend beschrieben werden.

Die im Elementenmodul definierten Elementklassen sind hierarchisch geordnet. Die Klassen stehen in einer als "Ableitung" bezeichneten Beziehung zueinander. Ableitung bedeutet, dass einer aus einer Basisklasse abgeleiteten Klasse alle Datenfelder (im folgenden als "Attribute" bezeichnet) und alle Programmfunktionen (im folgenden als "Methoden" bezeichnet) dieser Ba-

sisklasse zur Verfügung stehen. Darüber hinaus ist die Ableitung in der Lage, die Methoden zu verändern, das heißt, das in der Methode implementierte Verhalten zu modifizieren.

Wie die Elemente selbst, so sind auch die Elementklassen nach steuernden und speichernden Elementklassen unterschieden. Die Elemente basieren jedoch auf einer gemeinsamen Basisklasse.

Diese Klasse verfügt über zwei Methoden:

- Der Konstruktor: Eine Methode, die die Attribute eines neu erzeugten Elementobjektes initialisiert. Diese Methode wird unmittelbar nach der Erzeugung aufgerufen.
- GetValue: Die Methode wird von einem verbundenen Elementobjekt aufgerufen, um den Wert eines abgehenden Datenflusses abzufragen. Der Datenfluss trägt eine zeitkontinuierliche Größe.
- HasEvent: Die Methode wird von einem verbundenen Elementobjekt aufgerufen, um das Vorliegen eines Ereignisses an einem abgehenden Datenfluss abzufragen. Der Datenfluss trägt eine zeitdiskrete Größe.

Im folgenden werden steuernde und speichernde Elementklassen beschrieben.

Alle steuernden Elementklassen basieren auf einer gemeinsamen Basisklasse für steuernde Elementklassen. Diese Klasse enthält Attribute, die für die Ausführung eines steuernden Elementobjekts notwendig sind. Hierzu zählen insbesondere Datenfelder für die Stelle (Variable), die anzeigt, ob das steuernde Elementobjekt aktiv ist.

Die Klasse enthält darüber hinaus Methoden, die für die Erzeugung und die Ausführung der steuernden Elementobjekte notwendig sind. Die Methoden lassen sich in ein Steuerungsinterface und ein Prozessinterface aufgliedern.

Das Steuerungsinterface eines steuernden Elementobjekts gibt Auskunft über seinen Zustand und kann diesen beeinflussen. Dies sind insbesondere:

- **IsActive:** Diese Methode gibt Auskunft darüber, ob das betreffende Elementmodul aktiv ist, das heißt, seine Funktion ausführt. Kennzeichen für Aktivität ist das mit der Variablen assoziierte Attribut.
- **ProcessMark:** Die Methode wird aufgerufen, wenn das Elementobjekt einen geänderten Wert der Variablen erhält. Die Variable kann ihren dabei sowohl derart ändern, dass sie dem aktiven als auch dem deaktivierten Zustand entspricht. Ändert die Variable ihren Wert in Richtung einer Aktivierung, so wird unterschieden:
 - Die Variable hatte vorher einen der Deaktivierung entsprechenden Wert. Die Variable wird dann entsprechend geändert und die Methode **OnStart** wird aufgerufen.
 - Die Variable hatte bereits einen der Aktivierung entsprechenden Wert. Es erfolgt keine Reaktion.Ändert die Variable ihren Wert in Richtung einer Deaktivierung, so wird unterschieden:
 - Die Variable hatte vorher einen der Aktivierung entsprechenden Wert. Dann wird der Wert der Variablen entsprechend geändert und die Methode **OnTerminate** wird aufgerufen. Ein entsprechendes externes Steuersignal wird an alle abgehenden Steuerausgänge weitergereicht.
 - Die Variable hatte vorher einen der Deaktivierung entsprechenden Wert. In diesem Fall wird das externe Steuersignal an alle abgehenden Steuerausgänge weitergereicht.
- **RequestControl:** Die Methode wird von einem nachfolgenden Elementobjekt verwendet, um einen Steuerausgang auf das Vorhandensein eines entsprechenden externen Steuersignals zu überwachen. Ist ein entsprechendes externes Steuersignal vorhanden, so wird dieses externe Steuersignal dem Aufrufer übergeben und am betreffenden Ausgang gelöscht.
- **OnStart:** Die Methode wird jedes Mal aufgerufen, wenn das Elementobjekt vom inaktiven in den aktiven Zustand über-

geht, und initialisiert alle zur Erfüllung der Funktion notwendigen Datenfelder.

- OnTerminate: Die Methode wird jedes Mal aufgerufen, wenn das Elementobjekt vom aktiven in den inaktiven Zustand übergeht.

Das Prozessinterface dient der Ausführung der vom Elementobjekt implementierten Funktion. Es verfügt im speziellen über folgende Methoden:

- Update: Diese Methode wird zyklisch von einem Managementobjekt aufgerufen. Die überwacht und steuert den Zustand des Elementobjekts, indem sie sich der Methoden des Steuerungsinterface bedient. Der in Update implementierte Algorithmus arbeitet folgendermaßen:
- Der Steuereingang wird auf vorhandene externe Steuersignale mit dem ersten Signalpegel überwacht. Liegt ein entsprechendes Steuersignal vor, so wird dieses mit Process-Mark verarbeitet. Daraufhin kann sich er in der Variablen gespeicherte Wert ändern.
- Ist das Elementobjekt nach obigem Schritt aktiv, so wird die nachfolgend beschriebene Methode Process aufgerufen. In dieser kann sich der Zustand erneut ändern.
- Process: Diese Methode wird von Update aufgerufen, solange ein Elementobjekt als aktiv gekennzeichnet ist. Innerhalb dieser Methode wird die Aufgabe des Elementobjekts erfüllt. Um den Einsatz in einer echtzeitfähigen Umgebung zu ermöglichen, muss diese Methode mit einem Minimum an Rechenzeit implementiert werden. Üblicherweise wird die Methode so implementiert, dass nur jeweils ein kurzer Rechenschritt innerhalb der Aufgabe ausgeführt wird. Hierzu werden komplexe Aufgaben als getakteter Zustandautomat implementiert.

Alle speichernden Elementklassen basieren auf einer gemeinsamen Basisklasse für speichernde Elementklassen. Im Gegensatz zu steuernden Elementklassen verfügen speichernde Elementklassen

nur über ein Steuerungsinterface. Dieses besteht lediglich aus der Neudefinition folgender geerbter Methoden:

- Der Konstruktor: Eine Methode, die die Attribute eines neu erzeugten Elementobjektes initialisiert. Diese Methode wird unmittelbar nach der Erzeugung aufgerufen.
- GetValue: Die Methode wird von einem verbundenen Elementobjekt aufgerufen, um den Wert eines abgehenden Datenflusses abzufragen. Der Datenfluss träge eine zeitkontinuierliche Größe.
- HasEvent: Die Methode wird von einem verbundenen Elementobjekt aufgerufen, um das Vorliegen eines Ereignisses an einem abgehenden Datenausgang abzufragen. Über den Datenausgang wird ein zeitdiskrete Größe ausgegeben.

Weiterhin wird genau eine Managementklasse definiert. Diese deklariert als Attribute folgende Listen:

- Liste aller gegenwärtig verfügbaren Elementobjekte. Diese Liste stellt das momentan geladene Testprogramm dar.
- Liste aller gegenwärtig aktiven Elementobjekte. Diese Liste stellt die aktiven Elemente des Testprogramms dar.

Diese Listen sind in einem Managementobjekt mit Elementobjekten bedatet. Die Bedatung der Liste der gegenwärtig verfügbaren Elementobjekte wird als "Programmierung der Testmaschine" bezeichnet. Die Programmierung erfolgt über entsprechende Methoden der Managementklasse. Die vollständig bedatete List enthält somit das Testprogramm.

Ist die Testmaschine programmiert, so kann das Programm ausgeführt werden. Hierzu wird eine entsprechende Methode des Managementobjekts aufgerufen. Diese beginnt die Ausführung mit dem Start des ersten Elementobjekts in der Liste aller gegenwärtig verfügbaren Elementobjekte. Dies geschieht durch Übernahme in die Liste der aktiven Elemente. Das Elementobjekt ist nachfolgend aktiv und führt seine Aufgabe aus. Die von ihm abhängigen Elementobjekte beobachten dessen Steuerausgänge und aktivieren

sich entsprechend. Das Programm ist fertig abgearbeitet, wenn kein Elementobjekt mehr aktiv ist und die Liste der aktiven Elemente mithin leer ist.

Damit ein Managementobjekt in einer echtzeitfähigen Umgebung einsetzbar ist, verfügt es über eine zyklisch aufzurufende Update-Routine. Diese verarbeitet jeweils einmal pro Zyklus alle aktiven Elementobjekte.

Folgende Methoden ergeben sich für die Managementklasse:

- Construct: Der Konstruktor initialisiert alle Felder des unmittelbar zuvor erzeugten Managementobjektes.
- Init: Die Methode löscht das programm und bereitet das Managementobjekt auf eine erneute Programmierung vor.
- AddElement: Ein neues Elementobjekt wird der Liste der verfügbaren Elementobjekte hinzugefügt.
- Update: Es wird ein Rechenschritt in allen aktiven Elementobjekten ausgeführt.
- Start: Das Programm wird gestartet-
- Terminate: Das Programm wird beendet.

Nachfolgend soll nochmals ein Beispiel erläutert werden. Es soll dabei ein Signal S, das das Testsystem bereitstellt, überwacht werden. Das Signal S soll innerhalb von 500 ms den Wert "1" annehmen. Zeigt S dieses Verhalten, so ist der Test erfolgreich verlaufen und das Ergebnis soll dann "gut" sein. Laufen die 500 ms ab, ohne dass S den Wert 1 annimmt, so ist der Test nicht erfolgreich verlaufen. Das Ergebnis soll dann "schlecht" sein.

Dieser Test kann aus den vorstehend beschriebenen Elementen wird in Figur 22 dargestellt realisiert werden.

Beginnend mit dem Start-Element 2201 verzweigt die Ausführung über das Fork-Element 2202 in drei parallel laufende Elemente a, b und c. Während das Vergleichselement 2203 den Signalwert S

mit dem Referenzwert 1 vergleicht und das Eintreffen dieser Bedingung erwartet, zählt das Timeout-Element 2204 die angegebene Zeit herunter. Das Guardian-Element 2205 überwacht gleichzeitig, ob zuerst die Bedingung wahr wird oder aber das Timeout-Element abläuft. Im ersten Fall setzt das Guardian-Element seinen Ausgang auf "gut, im anderen Fall auf "schlecht". Das Guardian-Element wird in beiden Fällen ein externes Steuersignal mit dem ersten Signalpegel an das nachfolgende Exit-Element 2206 ausgeben. Dieses wird daraufhin den gesamten Prozess stoppen, indem es ein externes Steuersignal mit dem zweiten Signalpegel an das Start-Element 2201 sendet. Mit der Variablen 2207 "Ergebnis" kann abgefragt werden, ob der Test erfolgreich verlief.

Die Deaktivierung des Timeout-Elements 2204 führt hier im Kontrollfluss zu keiner weiteren Ausführung, da das nachfolgende Terminator-Element 2208 das externe Steuersignal mit dem ersten Signalpegel vernichtet. Die Werte nicht angeschlossener abgehender Datenflüsse werden nicht benötigt und verfallen.

DaimlerChrysler AG
Stuttgart

FTP/E bk
27.07.2000

Patentansprüche

1. Element zur Durchführung und Dokumentation eines Programm- bzw. Testablaufs, wobei durch das Element bestimmte Funktionen ausführbar sind,

d a d u r c h g e k e n n z e i c h n e t ,
dass das Element (1) wenigstens einen Steuereingang (3) aufweist, dem ein externes Steuersignal (2) zuführbar ist, wobei in dem Element (1) eine Variable (10) abhängig von dem externen Steuersignal (2) und der Ausführung der Funktion durch das Element (1) veränderbar ist derart,

dass die Variable (10) einen von einem bestimmten Wert abweichenden Wert (bspw. ungleich "0") annimmt, wenn die Variable (10) den Wert (bspw. "0") hat und das externe Steuersignal einen ersten Signalpegel aufweist,

dass die Variable (10) den bestimmten Wert beibehält, wenn die Variable (10) den bestimmten Wert aufweist und erneut ein externes Steuersignal (2) angelegt wird, das den ersten Signalpegel aufweist.

2. Element nach Anspruch 1,

d a d u r c h g e k e n n z e i c h n e t ,
dass die Variable (10) zurückgesetzt wird, wenn die Ausführung der Funktion durch das Element (1) beendet wurde und dass bei einem Rücksetzen der Variablen (10) von dem Element (1) über wenigstens einen Steuerausgang (4) des Elements (1) ein externes Steuersignal (5) ausgegeben wird, das den ersten Signalpegel aufweist.

3. Element nach Anspruch 1 oder 2,

d a d u r c h g e k e n n z e i c h n e t ,

dass die Variable (10) den Wert "0" annimmt, wenn dem wenigstens einen Steuereingang (3) des Elements (1) ein externes Steuersignal (2) zugeführt wird, das einen zweiten Signalpegel aufweist, wobei dann über den wenigstens einen Steuerausgang (4) des Elements ein externes Steuersignal (5) ausgegeben wird, das den zweiten Signalpegel aufweist.

4. Element nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass das Element (1, 501) einen Steuereingang (3, 503) und mehrere Steuerausgänge (4, 504) aufweist, wobei das externe Steuersignal (5, 505) von dem Element (1, 501) über alle Steuerausgänge (5, 505) ausgegeben wird.

5. Element nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass das Element (1, 601) mehrere Steuereingänge (3, 603) aufweist, wobei ein externes Steuersignal (2, 602), das den ersten Signalpegel aufweist, über den wenigstens einen Steuerausgang (4, 604) ausgegeben wird, wenn an allen Steuereingängen (3, 603) ein Steuersignal (2, 602) anliegt, das dem ersten Signalpegel entspricht.

6. Element nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass das Element (1) mehrere Steuereingänge (3, 603) aufweist, wobei ein externes Steuersignal (2, 602), das den ersten Signalpegel aufweist, über den wenigstens einen Steuerausgang (4, 604) ausgegeben wird, wenn an wenigstens einem Steuereingang (3, 603) ein Steuersignal (2, 602) anliegt, das dem ersten Signalpegel entspricht.

7. Element nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass das Element (1, 701) einen Steuereingang (3, 703) und mehrere Steuerausgänge (4, 704) sowie wenigstens einen Dateneingang (7, 707) aufweist, wobei ein an dem Steuereingang (3, 703)

anliegendes externes Steuersignal (2, 702) mit dem ersten Signalpegel über den Steuerausgang (4, 704) ausgegeben wird, der abhängig von dem an dem wenigstens einen Dateneingang (7, 707) anliegenden Signal (8, 708) durch die Funktion des Elements (1, 701) ermittelt wird.

8. Element nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass das Element (1, 901) einen oder mehrere Dateneingänge (7, 907) sowie einen oder mehrere Datenausgänge (13, 913) aufweist, wobei die Funktion darin besteht, aus den an dem bzw. den Dateneingängen (7, 907) liegenden Dateneingangssignalen (8, 908) ein oder mehrere Datenausgangssignale (15, 915) zu bilden, das bzw. die über den bzw. die Datenausgänge (13, 913) ausgegeben werden.

9. Element nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, dass die Funktion des Elements (1, 1001, 1101, 1201) in einer Zeitmessung besteht.

10. Element nach Anspruch 9, dadurch gekennzeichnet, dass die Zeitmessung (1001) erfolgt, indem beginnend zu einem Zeitpunkt, der durch das Anliegen eines externen Steuersignals (1002) mit dem ersten Signalpegel oder durch ein bestimmtes Signal an einem Dateneingang bzw. eine Signalkombination an mehreren Dateneingängen des Elements (1001) definiert ist, eine bestimmte Zeitdauer gemessen wird, wobei am Ende der Zeitdauer über einen Steuerausgang (1004) des Elements (1001) ein externes Steuersignal (1005) mit dem ersten Signalpegel ausgegeben wird und/oder an einem oder mehreren Datenausgängen (1013) ein entsprechendes Datensignal (1015) ausgegeben wird.

11. Element nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet,

dass dem Element (1, 1701) über einen Steuereingang (3, 1703) ein externes Steuersignal (2, 1702) zuführbar ist, wobei über einen Datenausgang (1713) des Elements (1, 1701) ein zeitlicher Signalverlauf als Datenausgangssignal (1715) ausgegeben wird, wenn das externe Steuersignal (1705) den ersten Signalpegel aufweist.

1 / 12

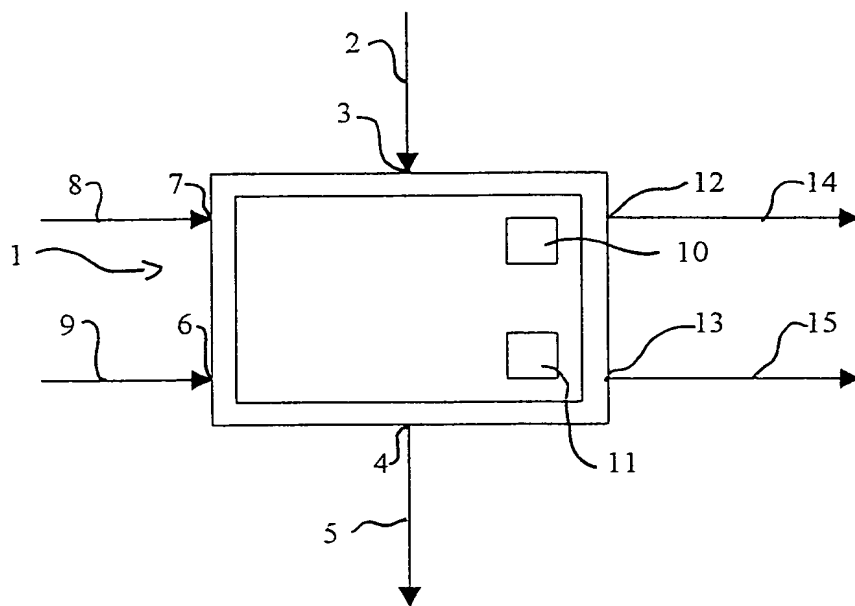


Fig. 1

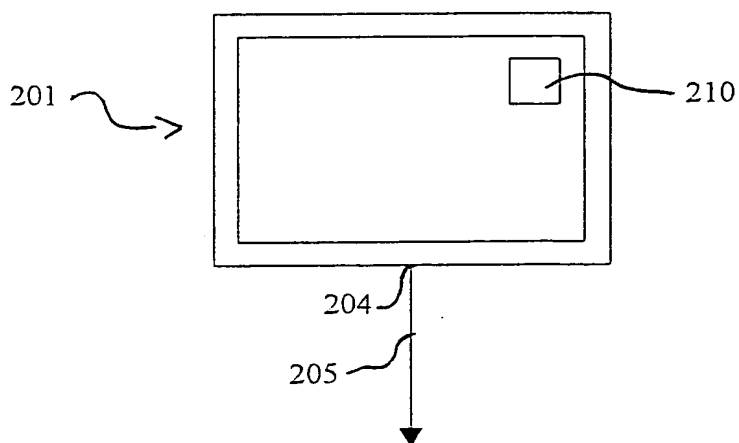


Fig. 2

2 / 12

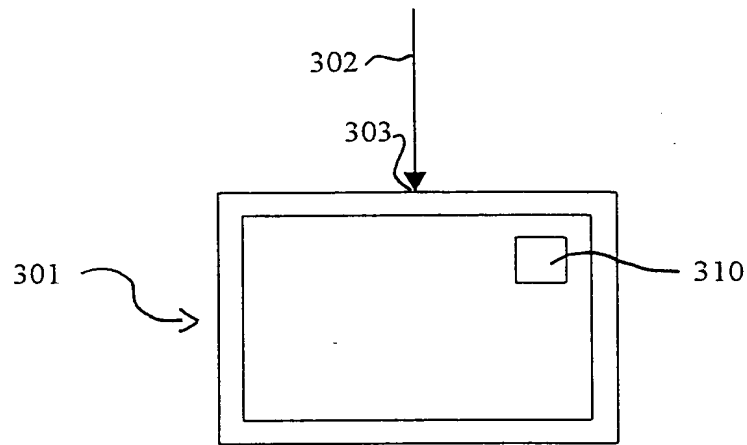


Fig. 3

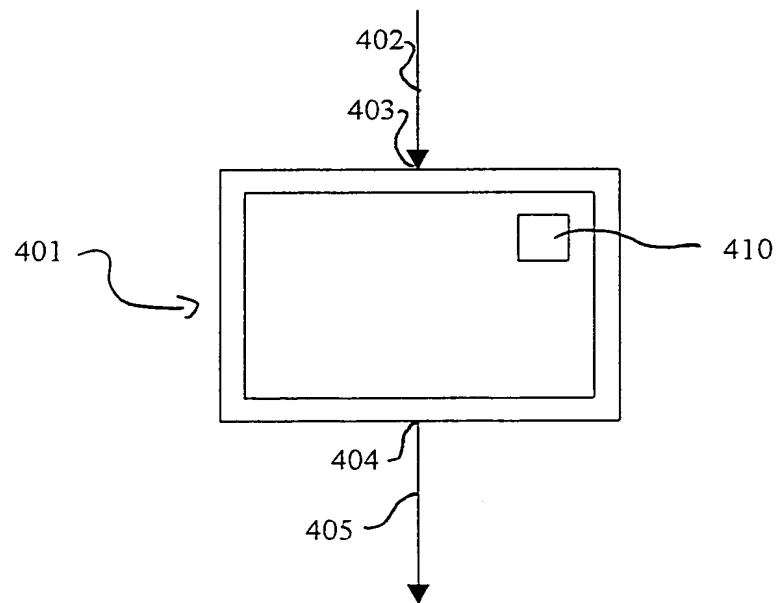


Fig. 4

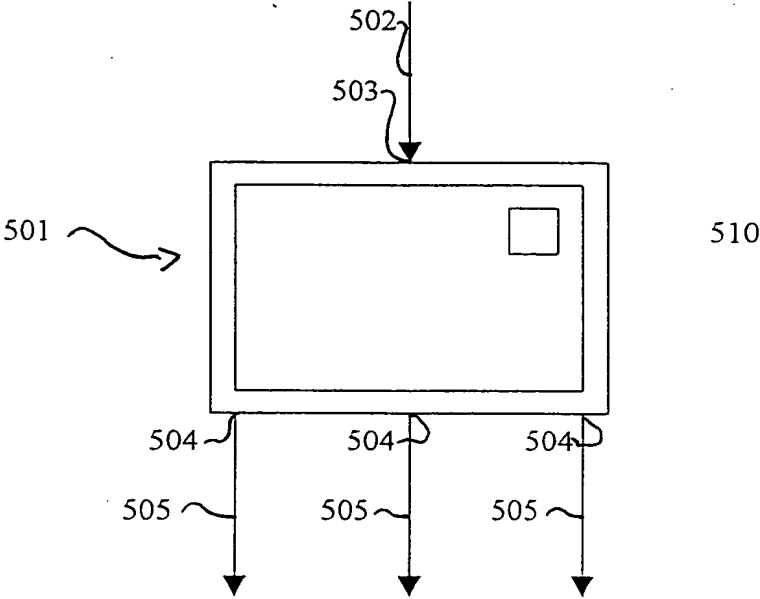


Fig. 5

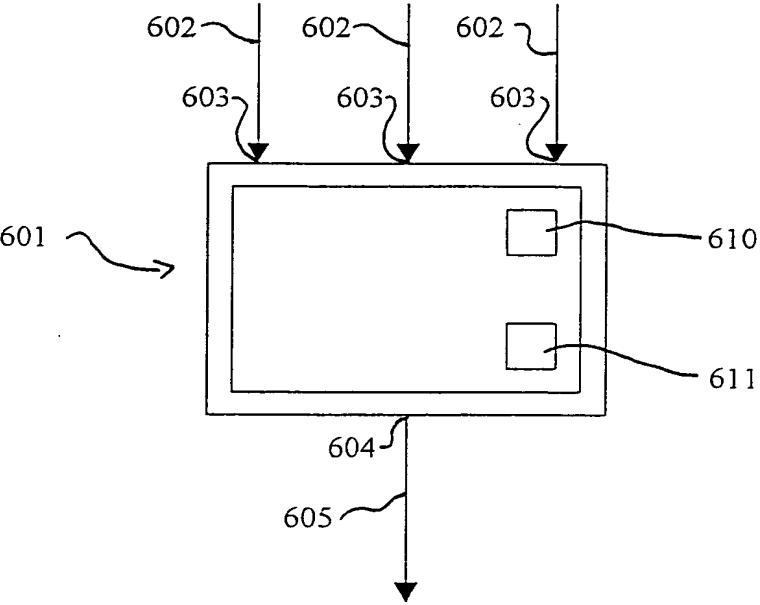


Fig. 6

4 / 12

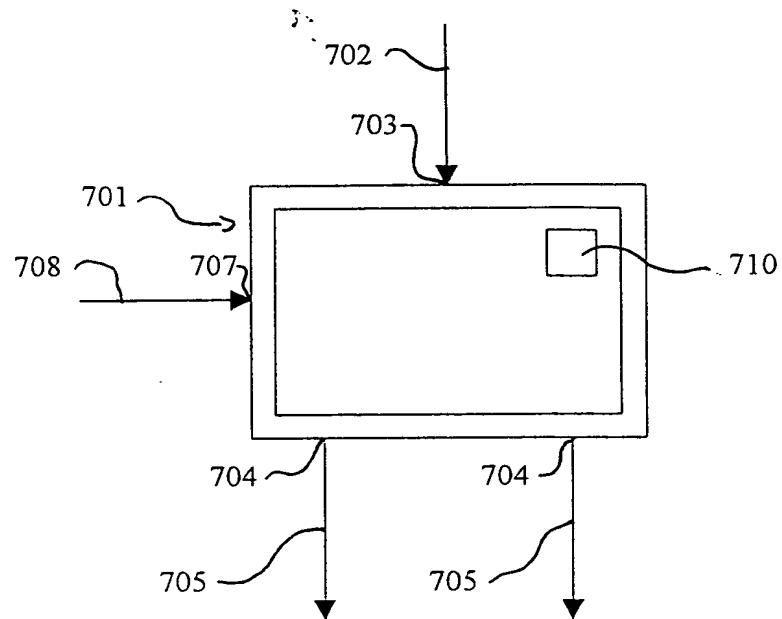


Fig. 7

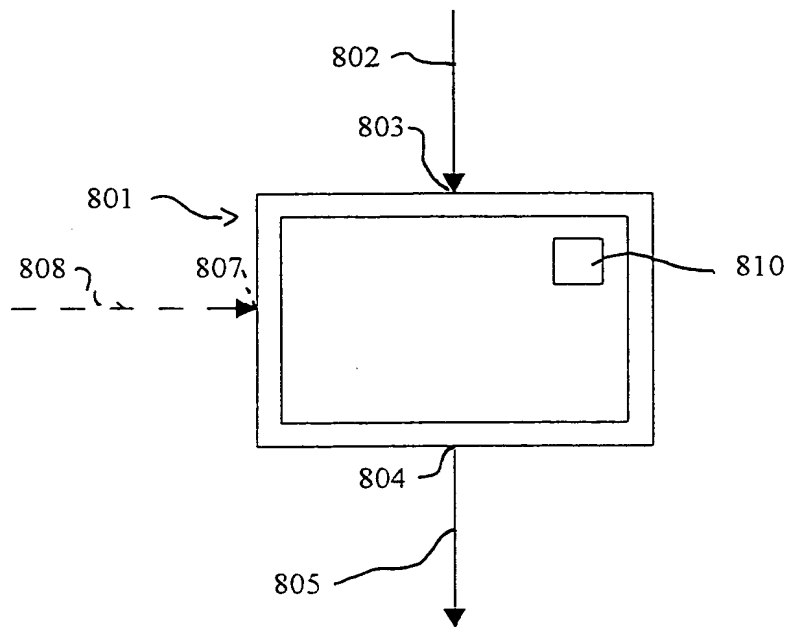


Fig. 8

5 / 12

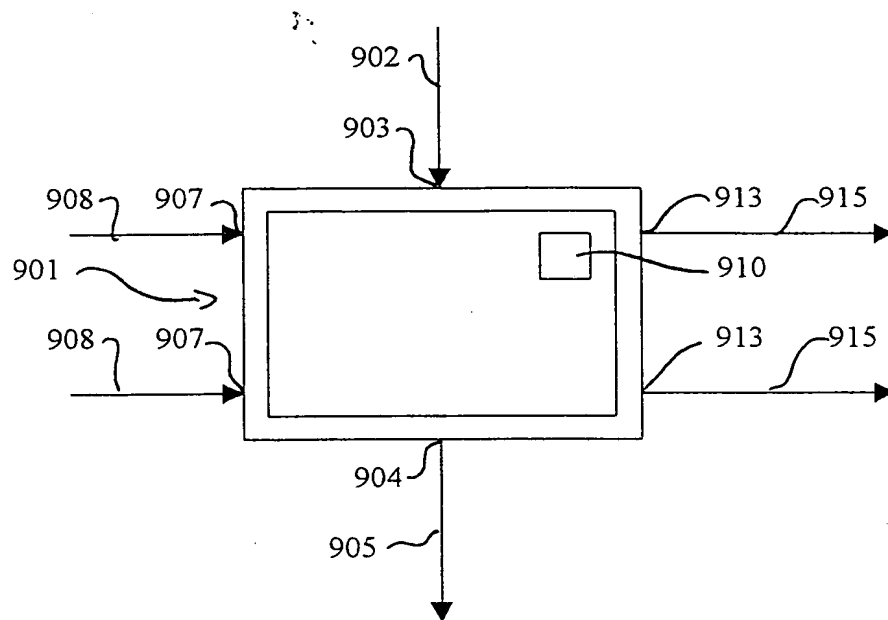


Fig. 9

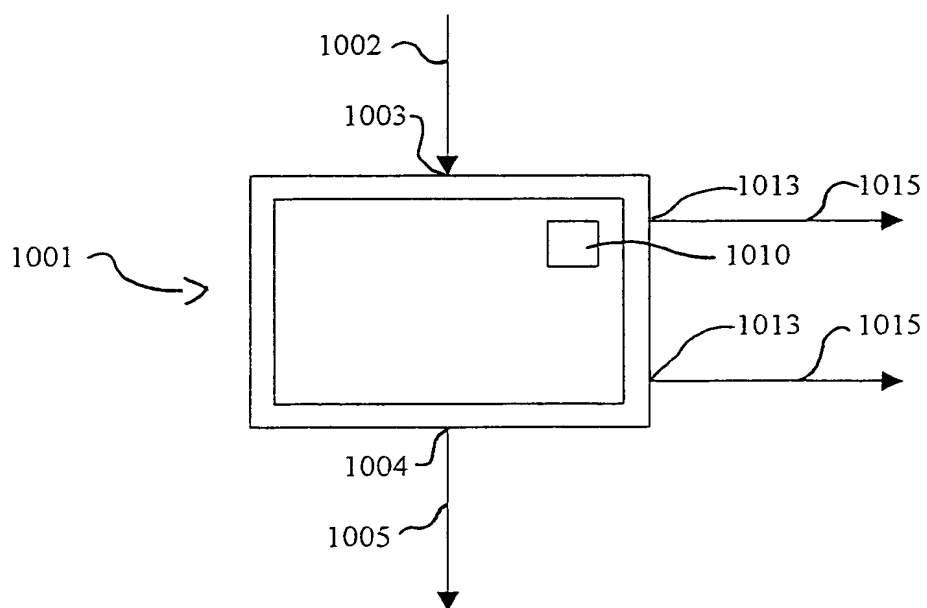


Fig. 10

6 / 12

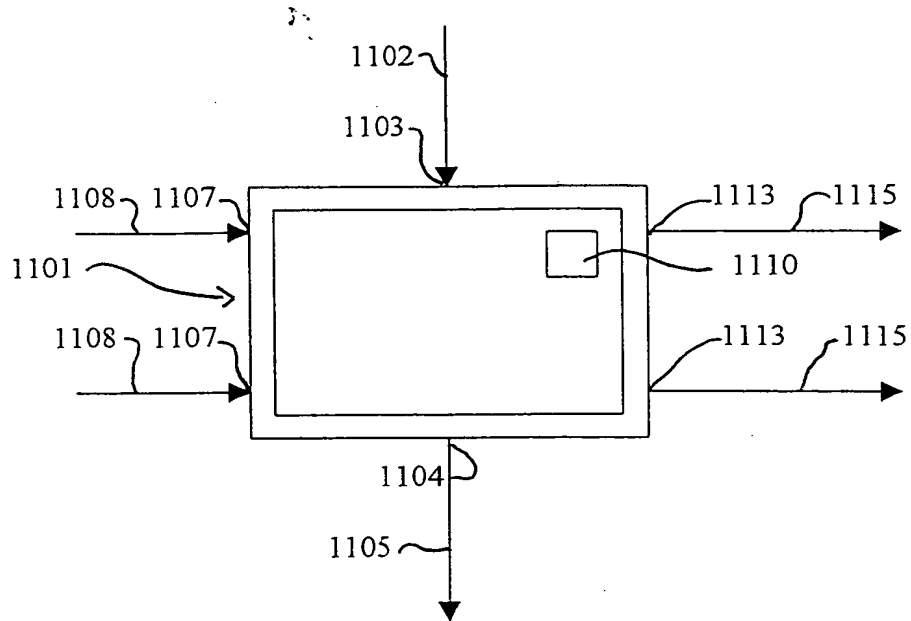


Fig. 11

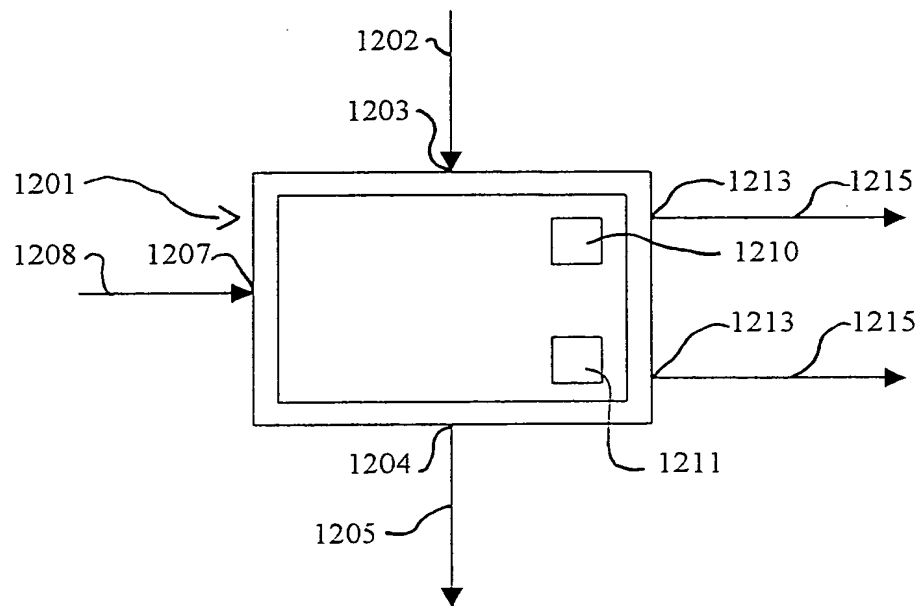


Fig. 12

7 / 12

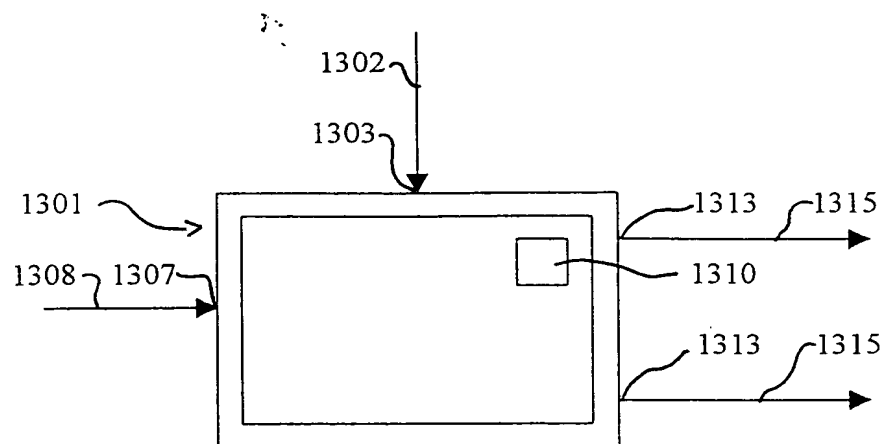


Fig. 13

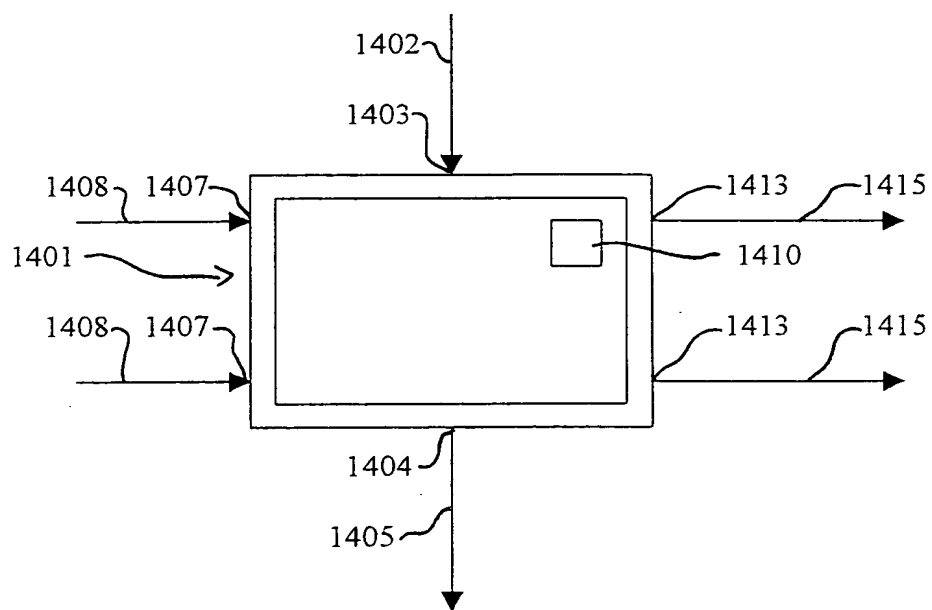


Fig. 14

8 / 12

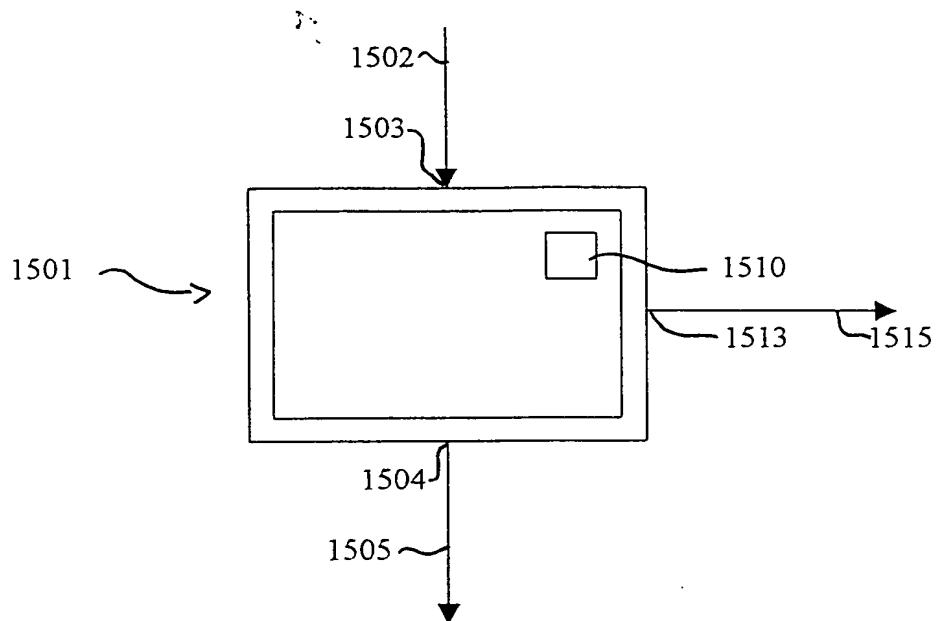


Fig. 15

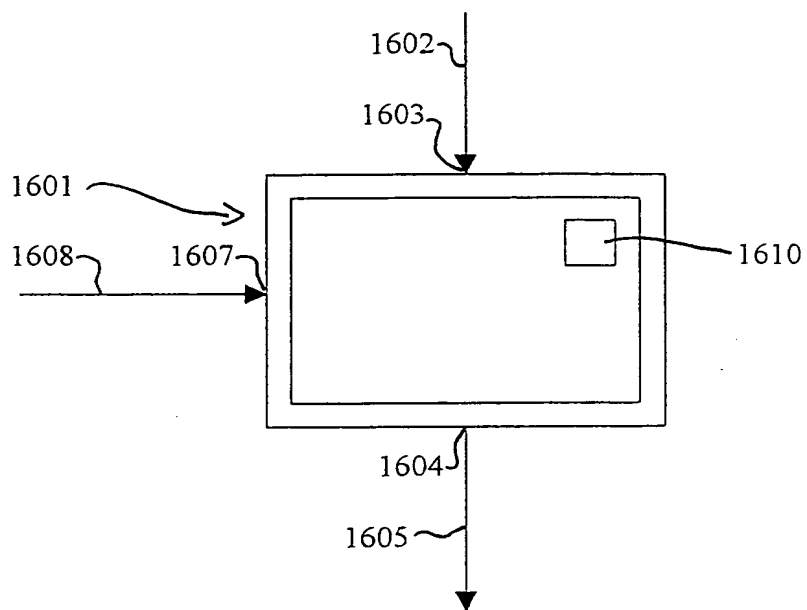


Fig. 16

9 / 12

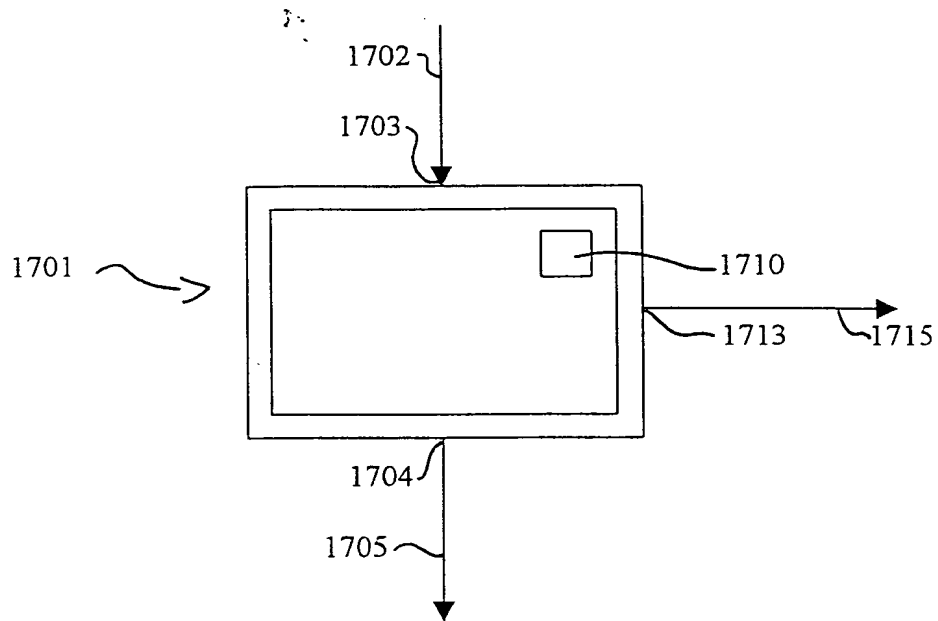


Fig. 17

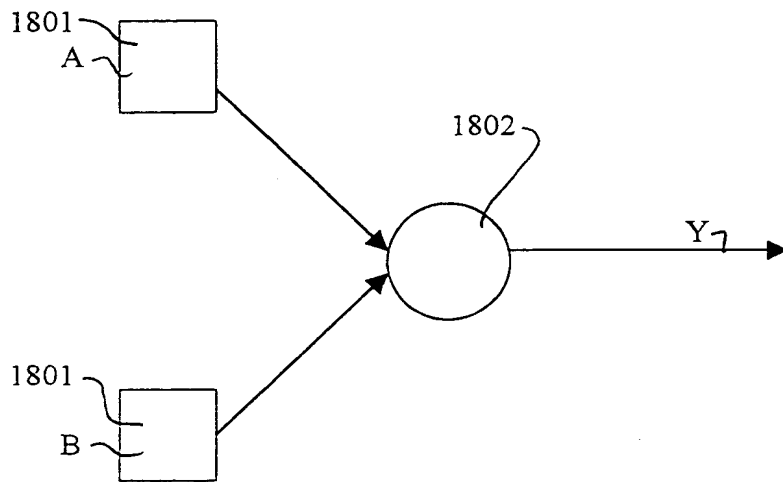


Fig. 18

10 / 12

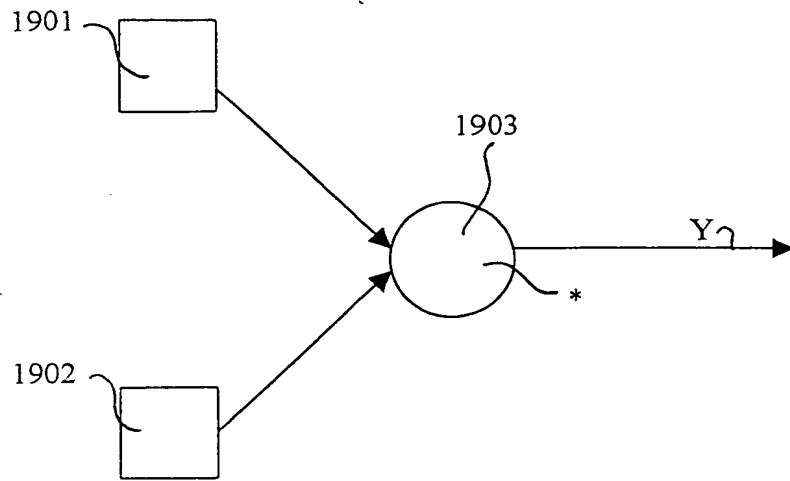


Fig. 19

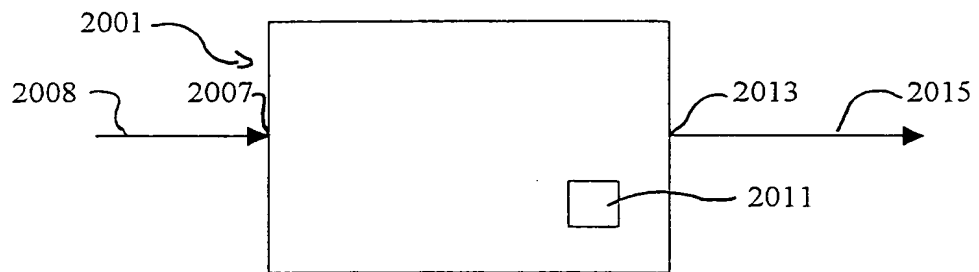


Fig. 20

11 / 12

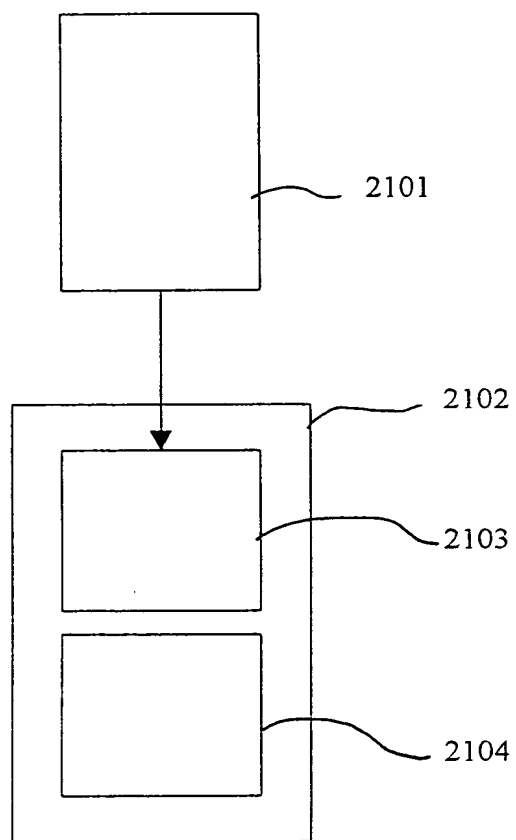


Fig. 21

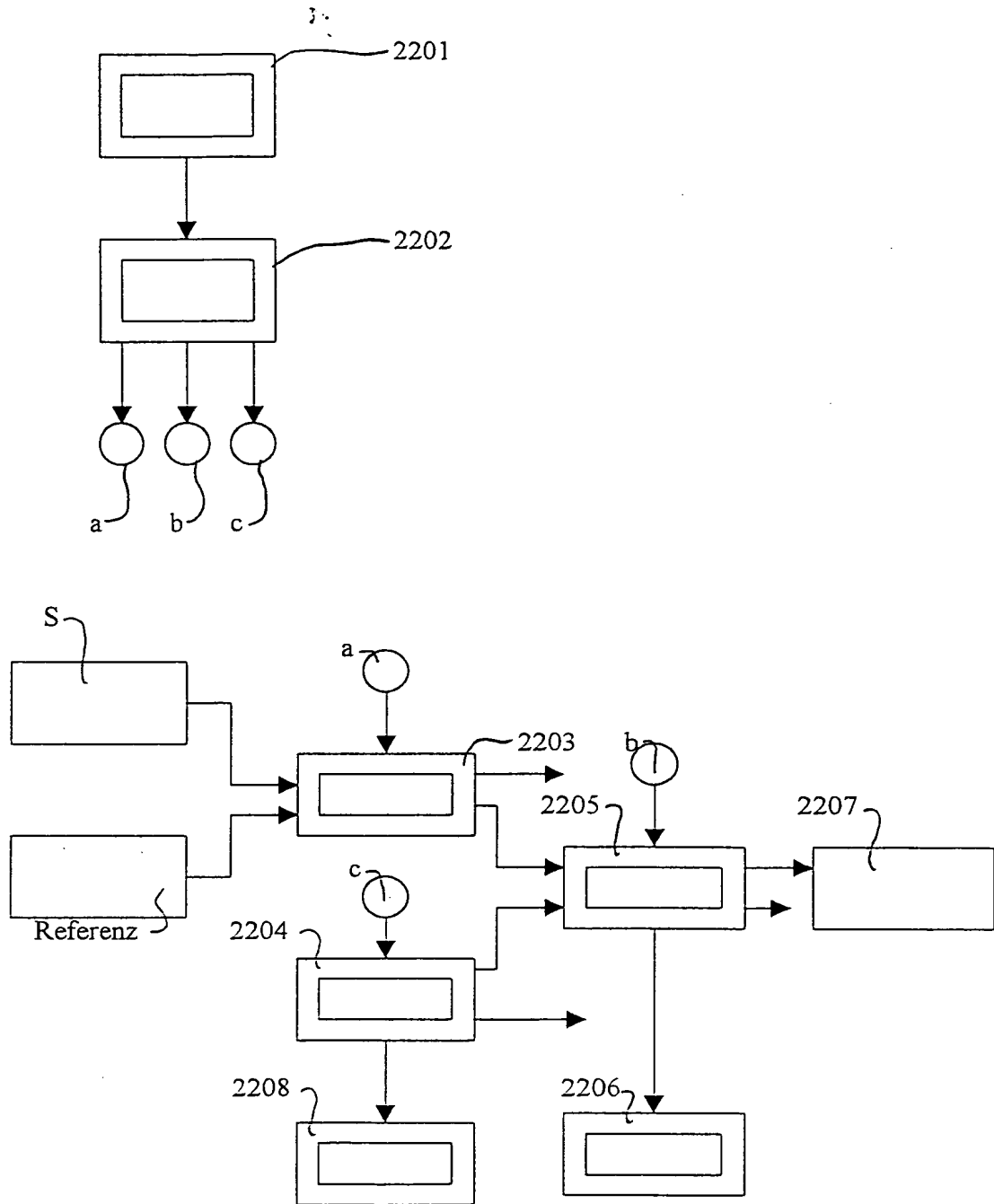


Fig. 22

DaimlerChrysler AG
Stuttgart

FTP/E bk
27.07.2000

Zusammenfassung

Die vorliegende Erfindung betrifft ein Element zur Durchführung und Dokumentation eines Programm- bzw. Testablaufs, wobei durch das Element bestimmte Funktionen ausführbar sind, wobei das Element wenigstens einen Steuereingang aufweist, dem ein externes Steuersignal zuführbar ist, wobei in dem Element eine Variable abhängig von dem externen Steuersignal und der Ausführung der Funktion durch das Element veränderbar ist derart, dass die Variable einen von einem bestimmten Wert abweichenden Wert (ungleich "0") annimmt, wenn die Variable den Wert ("0") hat und das externe Steuersignal einen ersten Signalpegel aufweist, und dass die Variable den bestimmten Wert beibehält, wenn die Variable den bestimmten Wert aufweist und erneut ein externes Steuersignal angelegt wird, das den ersten Signalpegel aufweist.

RECEIVED

AUG 07 2002

OFFICE OF PETITIONS